

51CTO | 开发频道 首页 Web 架构&设计 语言&工具 大数据

输入您要搜索的内容

用Web socket和Node.js实现HTML 5画布的实时绘图

Web套接字(Web socket)和画布(Canvas)是目前实施到浏览器中的两项超酷的特性。本教程将简要介绍这两者是如何工作的,并且制作了基于Node.js和Web套接字的实时画布。为了简洁起见,我将使用CoffeeScript来编写所有代码。如果你青睐用传统的JavaScript来编写,只要看一下相应的.js文件。出于同样的原因,我也弃用了CSS。

作者:布加迪编译 来源:51CTO | 2012-01-09 13:24

收藏 分享

Tech Neo技术沙龙 | 11月25号,九州云/ZStack与您一起探讨云时代网络边界管理实践

Web套接字(Web socket)和画布(Canvas)是目前实施到浏览器中的两项超酷的特性。本教程将简要介绍这两者是如何工作的,并且制作了基于Node.js和Web套接字的实时画布。为了简洁起见,我将使用CoffeeScript来编写所有代码。如果你青睐用传统的JavaScript来编写,只要看一下相应的.js文件。出于同样的原因,我也弃用了CSS。

从GitHub下载代码(<https://github.com/wesbos/websocket-canvas-draw>)



编辑推荐

热点

惹毛程序员的十件事！需求变更居然不是排第一！

头条

高性能Java持久化的14个技巧

介绍详细内容的屏幕播放速成教程

http://www.youtube.com/watch?feature=player_embedded&v=n7wQsLu_k00

跨设备/浏览器的兼容性

http://www.youtube.com/watch?v=oI9AahO9vDY&feature=player_embedded

服务器端

我们要做的第一件事就是建立一台Web套接字服务器。为此，我们要用到Node.js和模块Socket.io。Socket.io使得搭建和运行Web套接字服务器的任务超容易完成。它甚至为不支持原生Web套接字的浏览器提供了可退回到Flash的功能(Flash fallback)。在本教程中，我们将只使用支持元素的浏览器。

如果你之前还没有安装Socket.io，就要确保已安装，为此需要往你的终端中输入npm install socket.io。

现在，我们不妨建立Web套接字服务器。创建server.coffee文件，采用下列配置。

```
io = require('socket.io').listen(4000)
```

```
io.sockets.on 'connection', (socket) ->
```

编译你的coffeescript，重新回到你的终端，输入node server.js。现在你已有了一台通过端口4000运行的Web套接字服务器。

热点 号称世界最快句法分析器，Python高级自然语言处理库spaCy！

头条 跨界转行做编程的5大女神，新一代码农女神在谷歌做实习生！

头条 腾讯面试官送给准程序员的一些建议！

24H热文 一周话题 本月最赞

坐在马桶上看算法：快速排序

5个强大的Java分布式缓存框架推荐

Java程序员新手老手都离不开八大开发工具

二维码的生成细节和原理

Java 中常用缓存Cache机制的实现

我用Python爬了7W知乎用户信息，终于捕...

成为Java顶尖程序员，看这11本书就够了

挨踢部落坐诊第十一期：三千万数据如何做...



视频课程

+更多

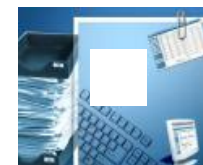
```
$ ls
server.coffee  server.js
$ node server.js
info - socket.io started
```

如果你进入到本地主机：4000，会看到下列结果：

客户端

首先，我们不妨立马让index.html文件搭建和运行起来。除了一些基本的标记外，我还加入了jQuery、现在从我们这个服务器提供的Socket.io JS文件、面向拖动事件的jQuery插件以及我们自己的scripts.js文件。

```
1. <!DOCTYPE HTML>
2. <html>
3. <head>
4.   <meta charset="UTF-8">
5.   <script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.6.2/jquery.mi
6.   <script type="text/javascript" src="js/jquery.event.drag-2.0.js"></script>
7.   <script src="http://localhost:4000/socket.io/socket.io.js"></script>
8.   <script type="text/javascript" src="scripts.js"></script>
9.   <link rel="stylesheet" href="style.css" />
10.
11.   <title>HTML5 Canvas + Node.JS Socket.io</title>
12. </head>
13. <body>
14.   <article><!--我们的画布将在此插入--></article>
15.
```



韩立刚Windows Server 2008
系统管理视频教程

讲师：韩立刚 244722人学习过



2017年软考系统分析师视频课程【综合知识】

讲师：小任老师 41564人学习过



SQL Server 2005微软数据库
管理精讲视频课程

讲师：韩立刚 132692人学习过

最新专题

+更多



未来即将“触脸可及”，人脸
识别技术大揭秘！

未来



关于智能运维的探索与实践

智能运维



智慧城市的背后是与前沿技术的
深度挖掘和利用

智慧城市

```

16. <!-- Scripts required -->
17. <script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.6.2/jquery.mi
18. <script type="text/javascript" src="js/jquery.event.drag-2.0.js"></script>
19. <script src="http://localhost:4000/socket.io/socket.io.js"></script>
20. <script type="text/javascript" src="scripts.js"></script>
21. </body>

```



HTML5游戏开发难点之效率、性能和加载量

HTML5游戏

精选博文 论坛热帖 下载排行

网络这些事儿-第一季

硬战--DC启动不了 (1/2)

Hyper-V实战：高可用性-配置群集

j2me手机游戏结构分析和实际小例子系

细说Java的接口 (一)

由于我们的服务器已搭建和运行起来，我们可以编写用来画到画布的一些代码。创建一个新文件，名为scripts.coffee。下面的所有代码都在App.init()方法里面执行，一旦jQuery文档准备就绪，我们就触发该方法。

创建我们的Canvas元素

#构建我们的应用程序，它有自己的命名空间

```

1. App = {}
2.
3. ###
4. 初始化
5. ###
6. App.init = ->
7.   App.canvas = document.createElement 'canvas' #创建<canvas>元素
8.   App.canvas.height = 400
9.   App.canvas.width = 800 #加大尺寸
10.  document.getElementsByTagName('article')[0].appendChild(App.canvas) #把它附加到DOM
11.
12.   App.ctx = App.canvas.getContext("2d") # 存储上下文
13.
14.   # 为我们的线条画设置一些参数
15.   App.ctx.fillStyle = "solid"

```

读书

+更多



C#入门经典(第3版)

本书将全面介绍C#编程的所有知识，共分为5篇：第1篇是C#语言：介绍了C#语言的所有内容，从基础知识到面向对象的技术，应有尽有。第2篇是Win...

```
16. App.ctx.strokeStyle = "#bada55"
17. App.ctx.lineWidth = 5
18. App.ctx.lineCap = "round"
19.
20. # 绘图函数
21. App.draw = (x,y,type) ->
22.   if type is "dragstart"
23.     App.ctx.beginPath()
24.     App.ctx.moveTo(x,y)
25.   else if type is "drag"
26.     App.ctx.lineTo(x,y)
27.     App.ctx.stroke()
28.   else
29.     App.ctx.closePath()
30.   return
```

**订阅51CTO邮刊**[点击这里查看样刊](#)[> 立即订阅](#)

画到画布函数

由于画到画布需要建立、移动和关闭路径，我创建了一个短小的函数，用来连接到jQuery拖动开始事件和拖动事件。

```
1. # 绘图函数
2. App.draw = (x,y,type) ->
3.   if type is "dragstart"
4.     App.ctx.beginPath()
5.     App.ctx.moveTo(x,y)
6.   else if type is "drag"
7.     App.ctx.lineTo(x,y)
8.     App.ctx.stroke()
9.   else
```

```
10.     App.ctx.closePath()
11.     return
```

设置客户端Web套接字

由于我们已将文件添加到<http://localhost:4000/socket.io/socket.io.js>，我们就能创建一个对象，以便通过该对象发送数据。只编写了几行代码，我们就创建了App.socket对象，并且绑定到名为“draw”的任何入站Web套接字事件。我们稍后对此会有介绍。

```
1. # 套接字!
2. App.socket = io.connect('http://localhost:4000')
3.
4. App.socket.on 'draw', (data) ->
5.     App.draw(data.x,data.y,data.type)
```

画布绘图事件

激动人心的一刻就出现在这里。现在我们想把几个事件绑定到我们的元素。其工作方式是，有人在画布上绘图时，我们立即使用draw()函数，画到当前画布上，并且使用socket.io的emit，通过Web套接字发送x和y坐标。稍后我们会看一下这个事件的服务器端部分，看看服务器如何将该数据发送到所有打开的窗口。

```
1. ###
2.     绘图事件
3. ###
4. $('canvas').live 'drag dragstart dragend', (e) ->
5.     type = e.handleObj.type
6.     offset = $(this).offset()
7.
```

```
8.     ee.offsetX = e.layerX - offset.left
9.     ee.offsetY = e.layerY - offset.top
10.    x = e.offsetX
11.    y = e.offsetY
12.    App.draw(x,y,type)
13.    App.socket.emit('drawClick', { x : x, y : y, type : type})
14.    return
```

回到服务器端

由于我们知道通过Web套接字发送x、y和事件类型，我们需要在服务器端进行一番处理。我们要做的就是，把该数据回过头来发送到浏览器打开的另外每个人。

我们更新后的server.coffee文件现在看来像这样。我们先等待连接事件，然后等待由浏览器发送的“drawClick”事件。发送完毕后，我们把数据发送到浏览器打开的另外每个人。我们之前编写的服务器端脚本然后就会绘制画布。

```
1.  io = require('socket.io').listen(4000)
2.
3.  io.sockets.on 'connection', (socket) ->
4.    socket.on 'drawClick', (data) ->
5.      socket.broadcast.emit 'draw',{ x : data.x, y : data.y, type: data.type}
6.      return
7.    return
```

你现在需要重启Web套接字服务器，因为我们已对它作了改动。按Control-c命令即可终止它，输入node server.js即可重启它。

开始绘图!

一旦你完全了解了这一切的工作原理，就可以在支持Web套接字和画布的任何Web浏览器(截至截稿时包括Chrome、Firefox、Safari、Opera和IE9)中打开你的index.html文件。访问<http://caniuse.com/#search=canvas>，即可获得更多的支持信息。

补充说明

由于这是一个非常基本的演示，存在几个局限性，不过再编写一点代码，就很容易克服。目前，画布只支持每次一个人绘图;如果两个或多个人同时绘图，画布就会被零星地绘制。另外，添加工具方面肯定也有很大的改进余地，比如缺少画笔、颜色、橡皮擦和PNG导出。

原文：<http://wesbos.com/html5-canvas-websockets-nodejs/>

【编辑推荐】

1. [走近Node.js的异步代码设计](#)
2. [使用Node.js开发多人玩的HTML 5游戏](#)
3. [Node.js提速指南](#)
4. [Node.js专区](#)
5. [使用JavaScript和Canvas写一个游戏框架](#)

【责任编辑：陈贻新 TEL：(010) 68476606】

点赞 0

HTML 5

分享:

大家都在看 猜你喜欢

51CTO旗下网站： 领先的IT技术网站 51CTO | 领先的中文存储媒体 WatchStor | 中国首个CIO网站 CIOage | 中国首家数字医疗网站 HC3i

Copyright©2005-2017 51CTO.COM 版权所有 未经许可 请勿转载